

Prof. Dr.-Ing. Jürgen Teich  
Lehrstuhl für Informatik 12  
(Hardware-Software-Co-Design)  
Friedrich-Alexander-Universität Erlangen-Nürnberg

## Klausur Grundlagen der Technischen Informatik

29. September 2017

Name	
Matrikelnummer	
Studienrichtung	

Aufgabe	1	2	3	4	5	$\Sigma$
Max. Punkte	15	15	20	15	15	80
Erreichte Punkte						
Note						

## Organisatorische Hinweise

Bitte sorgfältig lesen und die Kenntnisnahme durch Unterschrift bestätigen

---

- a) Bitte legen Sie Ihren Studentenausweis bereit.
  - b) Als Hilfsmittel sind nur Schreibmaterialien und ein beidseitig handbeschriebenes DIN A4-Blatt zugelassen.
  - c) Sie können bei der Aufsicht zusätzliche Bearbeitungsblätter anfordern.
  - d) Lösungen auf Schmierpapier werden nicht bewertet.
  - e) Bei mehreren präsentierten Lösungen wird die Aufgabe nicht bewertet. Streichen Sie daher bei Angabe mehrerer Lösungsansätze die nicht zu bewertenden Lösungen durch.
  - f) Unleserliches wird nicht bewertet.
- 

### Erklärung

- a) Im Falle einer während der Prüfung auftretenden Prüfungsunfähigkeit zeige ich dies sofort der Aufsicht an und befolge deren Anweisungen. Ich weiß, dass ich die volle Beweislast trage. Ich lasse mir das Formular des Prüfungsamts, das für diese Fälle vorgesehen ist, aushändigen und verfare nach den dort niedergelegten Richtlinien.
- b) Ich weiß, dass im Falle des Täuschungsversuchs oder der Benutzung unerlaubter Hilfsmittel („Unterschleif“) der Prüfungsausschuss die Entscheidung treffen kann, die betroffene Prüfungsleistung als mit „nicht ausreichend“ bewertet gelten zu lassen.
- c) Ich habe die obigen Hinweise zur Kenntnis genommen.

Erlangen, den .....

Unterschrift

Kopiervorlage: nur für Fachschaften

**Aufgabe 1 (Zahlensysteme)**

(15 Punkte)

a) Wie lautet der Wertebereich einer  $n$ -stelligen vorzeichenlosen Hexadezimalzahl? (1 Punkt)

b) Was sind die minimal und maximal möglichen Ergebnisse bei einer Multiplikation einer  $n$ -stelligen Binärzahl im Zweierkomplement mit einer  $m$ -stelligen Binärzahl im Zweierkomplement? (3 Punkte)

c) Addieren Sie die als BCDs codierten Zahlen  $(0110\ 0100\ 1000)_{\text{BCD}}$  und  $(0010\ 0101\ 1001)_{\text{BCD}}$ . Die Addition ist im Binärformat durchzuführen. Berücksichtigen Sie dabei die eventuell entstehenden Pseudotetrade. (3 Punkte)

d) Beantworten Sie folgende Auswahlfragen. Jede richtige Antwort gibt einen Punkt, jede falsche Antwort führt zum Abzug eines Punktes, nicht beantwortete Fragen werden nicht gewertet, weniger als null Punkte sind nicht möglich. (3 Punkte)

1. Mit Invertern und Oder-Gattern sowie den Konstanten **wahr** und **falsch** lässt sich jede beliebige Schaltfunktion darstellen.  wahr  falsch
2. Im Allgemeinen benötigt die Codierung einer vorzeichenlosen Dezimalzahl im BCD-Format mehr Bits als die Codierung als Gray-Code.  wahr  falsch
3. Die Zahl  $(-3,1)_{10}$  lässt sich ohne Rundungsfehler in die Gleitkommadarstellung nach IEEE-Standard 754 umwandeln.  wahr  falsch

- e) In dieser Aufgabe soll mit 16-Bit-Gleitkommazahlen gearbeitet werden, die analog zum IEEE-Standard 754 gebildet werden. Das Format einer Gleitkommazahl sieht dabei wie folgt aus:  
*Vorzeichen (1 Bit), Exponent (6 Bit), Mantisse (9 Bit)*

<i>V</i>	<i>E</i>	<i>M</i>
15	14 9	8 0

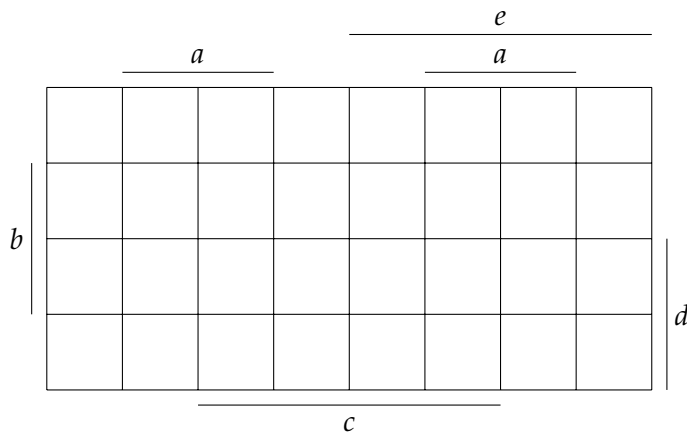
Führen Sie nun die Addition der beiden in diesem Format dargestellten Gleitkommazahlen 1 000111 000000100 und 0 000001 001111111 unter Beibehaltung des Gleitkommaformats durch, und geben Sie das Ergebnis im gleichen Format an. (5 Punkte)

Kopiervorlage: nur für Fachschaften

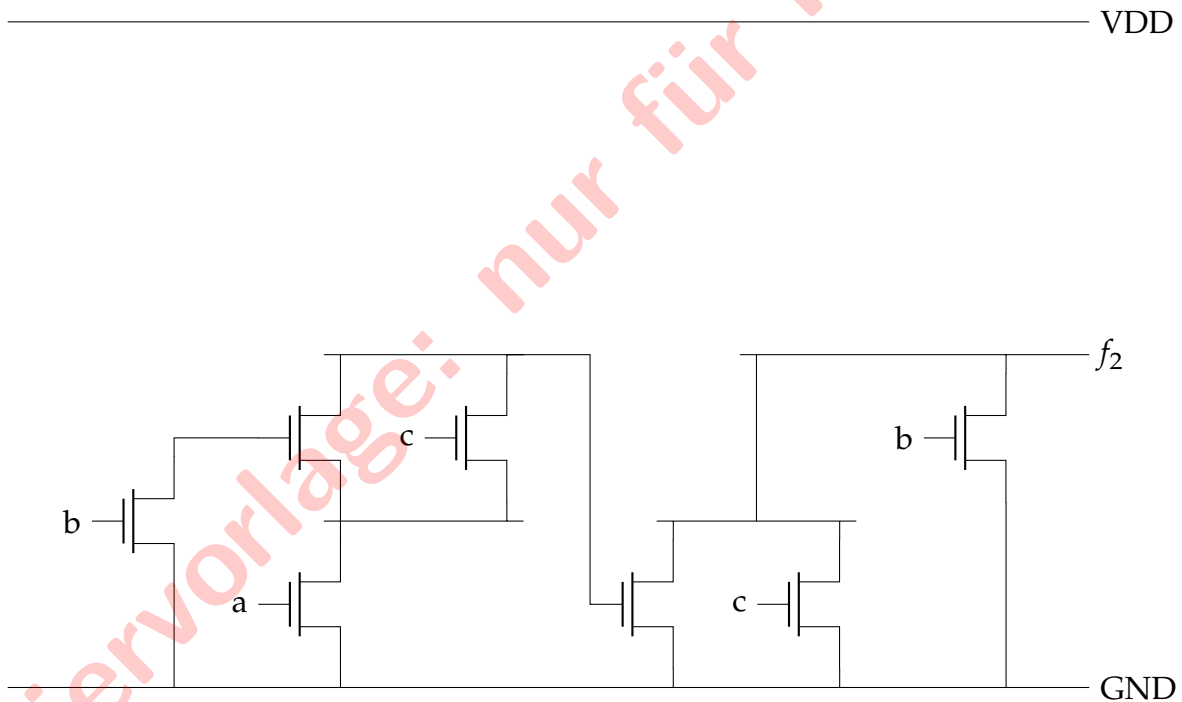
**Aufgabe 2 (Minimierung von Schaltfunktionen)**

(15 Punkte)

- a) Gegeben sei die Schaltfunktion  $f_1(e, d, c, b, a) = \bar{e}cb\bar{a} + edcb\bar{a} + \bar{c}b\bar{a} + \bar{d}\bar{c}b\bar{a} + \bar{e}\bar{b}\bar{a}$ . Bilden Sie mit Hilfe des gegebenen Symmetriediagramms eine DMF von  $f_1$ . (4 Punkte)



- b) Sei das Pull-Down-Netzwerk (PDN) einer CMOS-Schaltung der Schaltfunktion  $f_2(c, b, a)$  gegeben:



1. Geben Sie einen schaltalgebraischen Ausdruck für das PDN an. (2 Punkte)

2. Ergänzen Sie die CMOS-Schaltung durch das entsprechende Pull-Up-Netzwerk (PUN). (2 Punkte)

3. Geben Sie einen schaltalgebraischen Ausdruck für die Schaltfunktion  $f_2$  an. (1 Punkt)

4. Wie viele Transistoren benötigt die CMOS-Schaltung der Schaltfunktion  $f_2$  minimal? (1 Punkt)

c) Sei die Schaltfunktion  $f_3(e, d, c, b, a)$  mit Primimplikanten  $PI$  und Einsstellen  $E$  gegeben:

$$PI = \{\bar{e}\bar{c}\bar{a}, \bar{e}d\bar{a}, dc\bar{a}, b\bar{a}, cb, ec\}$$

$$E = \{\bar{e}d\bar{c}\bar{b}\bar{a}, \bar{e}d\bar{c}b\bar{a}, \bar{e}dcb\bar{a}, \bar{e}dcb\bar{a}, edcb\bar{a}\}$$

Bestimmen Sie mit Hilfe des Petrick-Verfahrens alle DMFs von  $f_3$ . (5 Punkte)

k	PI	j					$p_i$	$c_i$
		8	10	12	14	30		
0	$\bar{e}\bar{c}\bar{a}$						A	3
1	$\bar{e}d\bar{a}$						B	3
2	$dc\bar{a}$						C	3
3	$b\bar{a}$						D	2
4	$cb$						E	2
5	$ec$						F	2

Kopiervorlage: nur für Fachschaften

**Aufgabe 3 (Automaten)**

(20 Punkte)

- a) Im Folgenden soll ein Automat zur Überprüfung einer aus drei Ziffern bestehenden PIN entworfen werden. Hierfür verfügt der Automat über vier verschiedene Eingaben, die jedoch nicht gleichzeitig anliegen können: Die bisherige PIN-Eingabe wird zurückgesetzt (Eingabe *IReset*); die eingegebene Ziffer ist korrekt (Eingabe *ICorrect*); die eingegebene Ziffer ist falsch (Eingabe *IWrong*); die maximale Zeitspanne zur Eingabe einer weiteren Ziffer der PIN wurde überschritten bzw. das Ende einer Eingabesperre aufgrund einer falsch eingegebenen Ziffer wurde erreicht (Eingabe *ITimeout*).

Das Verhalten ist wie folgt spezifiziert: Befindet sich der Automat im Startzustand, sollen die Eingaben *IReset* und *ITimeout* ignoriert werden (Ausgabe *Idle*). Sobald die erste Ziffer korrekt eingegeben wurde (Eingabe *ICorrect*), soll dies mit der Ausgabe *Correct* bestätigt werden. Handelt es sich jedoch um eine falsche Ziffer (Eingabe *IWrong*), soll eine Eingabesperre erfolgen (Ausgabe *Locked*), bei der alle Eingaben außer *ITimeout* ignoriert werden. Erst nachdem das Ende der Eingabesperre signalisiert wurde (Eingabe *ITimeout*), geht der Automat wieder in den Startzustand über (Ausgabe *Idle*) und die PIN-Eingabe kann erneut erfolgen. Bei der falschen Eingabe der zweiten oder dritten Ziffer soll der Automat analog verfahren. Der Automat geht auch dann in den Startzustand zurück (Ausgabe *Idle*), wenn nach korrekter Eingabe der ersten bzw. zweiten Ziffer die maximale Zeitspanne zur Eingabe einer weiteren Ziffer der PIN überschritten wurde (Eingabe *ITimeout*). Zudem soll durch die Eingabe *IReset* außerhalb der Eingabesperre stets die bisherige PIN-Eingabe zurückgesetzt werden und der Automat in den Startzustand zurückkehren (Ausgabe *Idle*). Wird hingegen die zweite und die dritte Ziffer korrekt eingegeben (jeweils Eingabe *ICorrect*), soll dies mit den Ausgaben *Correct* (zweite Ziffer korrekt) und *Access* (dritte Ziffer korrekt) bestätigt werden. Nach korrekter Eingabe der dreistelligen PIN sollen außer *IReset* alle Eingaben ignoriert werden (Ausgabe *Access*).

Die Ein- und Ausgaben sind wie folgt durch binäre Variablen codiert:

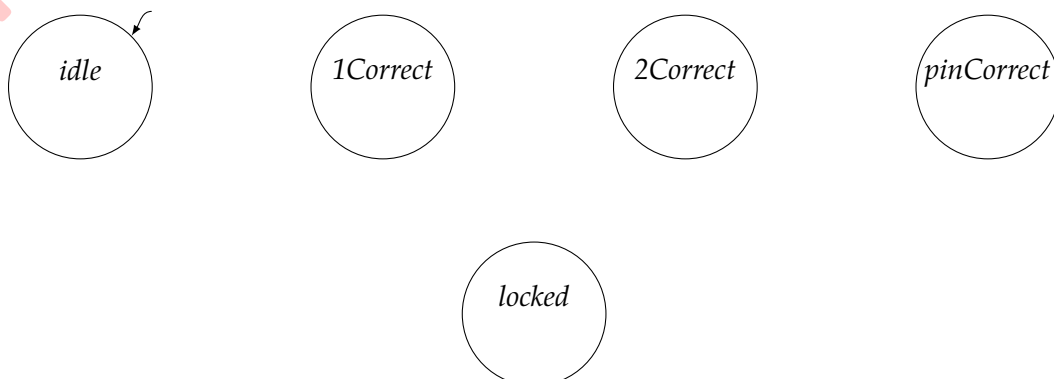
Eingabe	$i_1$	$i_0$
<i>IReset</i>	0	0
<i>ICorrect</i>	0	1
<i>IWrong</i>	1	0
<i>ITimeout</i>	1	1

Eingaben des Automaten

Ausgabe	$o_1$	$o_0$
<i>Idle</i>	0	0
<i>Correct</i>	0	1
<i>Access</i>	1	0
<i>Locked</i>	1	1

Ausgaben des Automaten

- Spezifizieren Sie den beschriebenen Automaten als Moore-Automat unter Verwendung der fünf Zustände *idle*, *1Correct* (erste Ziffer korrekt), *2Correct* (ersten zwei Ziffern korrekt), *pinCorrect* (alle drei Ziffern korrekt) und *locked* (Sperrzustand). Geben Sie den resultierenden Automatengraphen an. (5 Punkte)



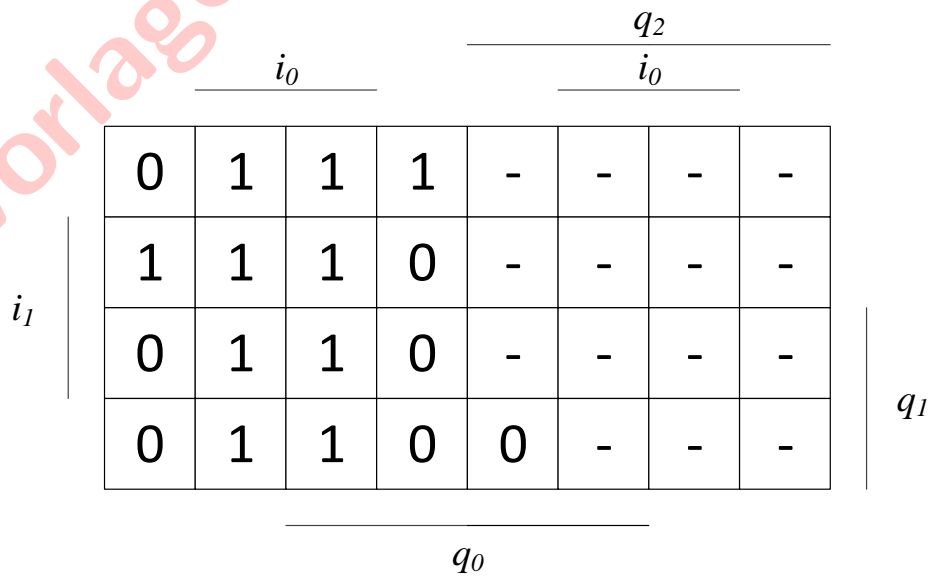


2. Vervollständigen Sie die nachfolgend gegebene Automatentafel unter Verwendung von taktflankengesteuerten D-, T- bzw. JK-Flipflops. (6 Punkte)

Zustandsname	Aktueller Zustand			Eingabe		Nachfolgezustand			Ansteuerfunktion				Ausgabe	
	$q_2$	$q_1$	$q_0$	$i_1$	$i_0$	$q'_2$	$q'_1$	$q'_0$	$D_2$	$T_1$	$J_0$	$K_0$	$o_1$	$o_0$
idle	0	0	0	0	0									
idle	0	0	0	0	1									
idle	0	0	0	1	0									
idle	0	0	0	1	1									
1Correct	0	0	1	0	0									
1Correct	0	0	1	0	1									
1Correct	0	0	1	1	0									
1Correct	0	0	1	1	1									
2Correct	0	1	0	0	0									
2Correct	0	1	0	0	1									
2Correct	0	1	0	1	0									
2Correct	0	1	0	1	1									
pinCorrect	0	1	1	0	0									
pinCorrect	0	1	1	0	1									
pinCorrect	0	1	1	1	0									
pinCorrect	0	1	1	1	1									
locked	1	0	0	0	0									
locked	1	0	0	0	1									
locked	1	0	0	1	0									
locked	1	0	0	1	1									

3. Nehmen Sie im Folgenden an, dass die Ansteuerfunktion  $J_0$  des JK-Flipflops im unten gegebenen Symmetriediagramm spezifiziert sei. Entwickeln Sie eine KMF der Ansteuerfunktion  $J_0$ , und geben Sie den resultierenden schaltalgebraischen Ausdruck an.

Hinweis: Die hier angegebene Ansteuerfunktion  $J_0$  entspricht nicht der Lösung von Teilaufgabe 2. (3 Punkte)



4. Zeichnen Sie ein Schaltwerk für den in Teilaufgabe 2 realisierten Moore-Automaten. Verwenden Sie dazu die in Teilaufgabe 3 bestimmte KMF der Ansteuerfunktion  $J_0$ , und gehen Sie davon aus, dass Sie die benötigten Ansteuerfunktionen  $D_2$ ,  $T_1$  und  $K_0$  als Eingangssignale zur Verfügung haben. Auf die Realisierung der Ausgabe des Automaten kann verzichtet werden. (3 Punkte)

- b) Beantworten Sie folgende Auswahlfragen. Jede richtige Antwort gibt einen Punkt, jede falsche Antwort führt zum Abzug eines Punktes, nicht beantwortete Fragen werden nicht gewertet, weniger als null Punkte sind nicht möglich. (3 Punkte)

1. Bei einem Medwedew-Automat ist die Ausgabe eine beliebige Funktion, abhängig alleine vom Zustand.  wahr  falsch
2. Die Realisierung eines Schaltwerkes mit 4 Eingängen, 5 Zuständen, 2 Ausgängen erfordert 2 Flipflops.  wahr  falsch
3. Mit einer Schaltung bestehend aus einem Inverter und einem Und-Gatter lassen sich positive Flanken erkennen.  wahr  falsch

**Aufgabe 4 (Codierung und Rechnerarithmetik)**

(15 Punkte)

a) Ein Kartendeck beim Schafkopf besteht aus 32 Karten mit jeweils 8 Karten derselben Farbe. Die Farben sind Eichel, Grün, Herz und Schellen. Bei einem Rufspiel sind 14 der 32 Karten Trumpf. Unter diesen Bedingungen werden nun Karten aus einem vollen Stapel gezogen.

1. Was ist der Informationsgehalt folgender Aussage  $A$ :

“Die erste gezogene Karte ist das Herz Ass”?

(1 Punkt)

2. Begründen Sie, ob der Informationsgehalt der Aussage  $B$  “Die erste gezogene Karte ist Trumpf” höher oder niedriger als der der Aussage  $A$  ist.

(1 Punkt)

3. Eine Quelle mit dem Alphabet

{“Die Karte ist Herz”,

“Die Karte ist Schellen”,

“Die Karte ist weder Herz noch Schellen”}

gibt Informationen über die erste gezogene Karte aus einem vollen Deck. Berechnen Sie die Entropie der Quelle.

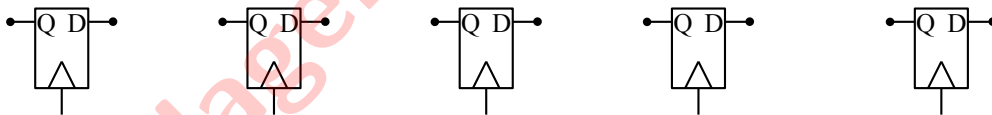
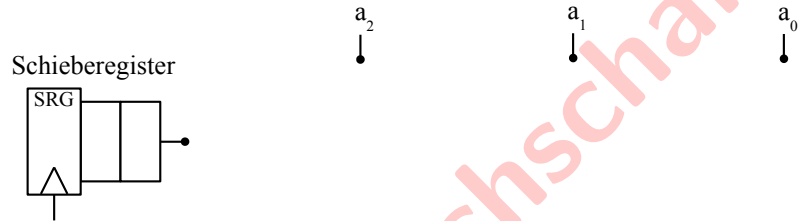
(3 Punkte)

b) Gegeben sei eine Quelle, deren Alphabet aus 4-Bit-Binärwörtern  $x_4x_3x_2x_1$  besteht. Geben Sie die Konstruktionsvorschrift für Prüfvektoren  $y_3y_2y_1$  entsprechend des Hamming-Code-Verfahrens an. Erstellen Sie dann das Codewort für 0011 entsprechend dieser Vorschrift.

(2 Punkte)

c) In dieser Aufgabe soll die Verzögerungszeit eines sequentiellen Multiplizierers ermittelt werden, der 3-Bit-Multiplikanten  $a_2a_1a_0$  mit 2-Bit-Multiplikatoren  $b_1b_0$  multipliziert und das Produkt  $p_4p_3p_2p_1p_0$  liefert.

- Ergänzen Sie das Schaltbild des Multiplizierers unter ausschließlicher Verwendung von Und-, Oder- und XOR-Gattern mit zwei Eingängen sowie Halbaddierern und Volladdierern. Kennzeichnen Sie außerdem, welchen Zellen des Schieberegisters anfangs  $b_1$  und  $b_0$  zugeordnet sind und in welchen D-Flipflops letztendlich die Bits  $p_4$  bis  $p_0$  des Produkts stehen. (4 Punkte)



- Nehmen Sie an, dass Und- und Oder-Gatter eine Verzögerungszeit von 2 ns, XOR-Gatter von 4 ns, Halbaddierer von 3 ns und Volladdierer von 7 ns haben. Geben Sie die Laufzeit zur Berechnung der kombinatorischen Logik an, und markieren Sie den längsten Pfad im Schaltbild. (1 Punkt)

Kopierverbot: nur für Fachschaften

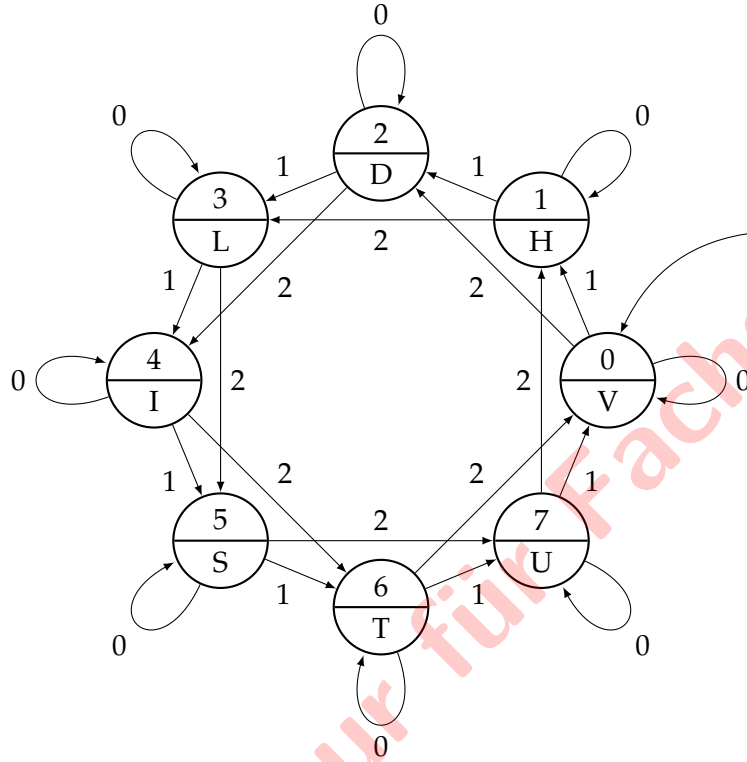
3. Geben Sie nun die Laufzeit der kombinatorischen Logik an, wenn weder Halbaddierer noch Volladdierer als funktionale Einheiten zur Verfügung stehen, sondern diese aus den verbleibenden Gattern (Und-, Oder- und XOR-Gatter) aufgebaut werden. Verwenden Sie die Verzögerungszeiten aus Teilaufgabe 2. (3 Punkte)

Kopiervorlage: nur für Fachschaften

## Aufgabe 5 (VHDL)

(15 Punkte)

Folgender Automatengraph beschreibt einen endlichen Automaten, der eingabeabhängig unterschiedliche Zeichenketten generiert. Jeder Zustand und jede mögliche Eingabe ist jeweils mit einer ganzen Zahl codiert, wobei die Topologie des Graphen einem chordalen Ring zweiten Grades entspricht.



Implementieren Sie den spezifizierten Automaten als *synchrone* Schaltung, die sich *asynchron* zurücksetzen lässt.

- a) Vervollständigen Sie die folgende *entity*-Deklaration. Sowohl Eingabe *input* als auch Ausgabe *output* sollen als *std\_logic\_vector* übermittelt werden. Die Ausgabe sei dabei mit ASCII codiert, es gibt also insgesamt 128 Zeichen. (2 Punkte)

```
entity automaton is
```

```
end automaton ;
```

- b) Wie müsste der Automaten geändert werden, damit die Zeichenkette „TUV“ ausgegeben werden kann? (1 Punkt)

- c) Vervollständigen Sie folgende Funktion, die die Zustandüberföhrungsfunktion  $\delta$  implementiert, also für eine gegebene Kombination von Zustand und Eingabe den nächsten Zustand zurückliefert. (2 Punkte)

```
function get_next_state(state: integer, input: integer)
    return integer is
```

```
begin
```

```
end function ;
```

- d) Vervollständigen Sie folgende Funktion, die die Ausgabefunktion  $\lambda$  implementiert, also für einen gegebenen Zustand die entsprechende Ausgabe zurückliefert. Im Falle eines ungültigen Zustands soll X zurückgegeben werden. *Hinweis: Die Ausgabe X ist nicht Teil des Automatengraphen, weil sie genau dann auftritt, wenn die Hardwarerealisierung aufgrund von Laufzeitfehlern in einem ungültigen Zustand ist.* (3 Punkte)

```
function get_output(state: integer) return character is
begin — Beispielsyntax: return 'P';
```

```
end function ;
```

- e) Vervollständigen Sie den Rumpf der folgenden architecture unter Verwendung der beiden obigen Funktionen. (4 Punkte)

```
architecture behavioral of automaton is
  signal state : integer := 0;
  signal input_int : integer := 0;
  signal output_char : character := ' ';
begin
  -- Ein-/Ausgabe in korrekte Typen umwandeln
  output <= character_to_std_logic_vector(output_char);
  input_int <= std_logic_vector_to_integer(input);
```

```
end architecture;
```

- f) Geben Sie die Werte der Signale d1, d2 und d3 nach dem nächsten Simulationsschritt des folgenden process an, wenn zu dessen Beginn gilt: d1="010", d2="110" und d3="011". Alle Signale sind vom Typ std\_logic\_vector(2 downto 0). (3 Punkte)

```
-- anfangs: d1="010", d2="110", d3="011"
process(d1, d2, d3)
  variable temp : std_logic_vector(2 downto 0) := "111";
begin
  temp := d2 and d3;
  d1 <= temp nand temp;
  d2 <= d1;
  temp := not temp;
  d3 <= d3 xor temp;
end process;
```